An Efficient Continuous Reverse k-Nearest Neighbor Query Processing Method with MapReduce

Jongtae Lim[1], Sunyong Park[1], He Li[1], Kyoungsoo Bok[1], Byoungyup Lee[2], Jaesoo Yoo[1]*
[1]Chungbuk National University, Korea, [2]Paichai University, Korea
E-mail : [1]{jtlim, sypark1992, lihe, ksbok, yjs}@chungbuk.ac.kr, [2]bylee@paichai.ac.kr

1. Introduction

With the development of location aware technologies and mobile devices, location-based services have been studied. To provide location-based services, various query types such as a point query, a range query, a k-Nearest Neighbor (kNN) query, a skyline query, and a top-k query have been presented. Especially, a kNN query is an important query type for location-based services. It returns k objects in nearest locations from the query. And one of variation query type of kNN, Reverse k-Nearest Neighbor (RkNN) query has been researched. RkNN query is the query that returns the results that contain the query in his kNN results.

Hadoop has been developed in order to process big data. Hadoop consists of a Hadoop distributed file system (HDFS) and MapReduce (MR). MR is the framework to provide the distributed data processing in big data environments. Recently, many researchers proposed methods for processing various query types with MR. One of the proposed methods, is a RkNN query processing method with MR. There are RkNN query processing methods with MR such as Voronoi-based method [1] and Lazy-MRRkNN [2]. Voronoi-based method [1] is the RkNN query processing method using voronoi-diagram. This method builds the voronoi-diagram in pre-processing step. When processing a query, this method executes the map processing using voronoi-diagram. However, this method requires high costs to build the voronoi-diagram. When data is modified, this method must rebuild the voronoi-diagram to process the query. Lazy-MRRkNN [2] is the RkNN query processing method using the feature of bisector. This method does not need the pre-processing. When processing the query, this method draws the bisector between the query and neighbor objects. This method prunes the objects that are not included in the RkNN query by using the bisector. However, this method also requires high costs to process the continuous query. The reason is that when objects are modified, it must redraw the bisectors. This method may spend additional costs because it must process a range query in order to draw the bisector in non-uniform objects environments.

In this paper, we propose an efficient continuous RkNN query processing method with MR to resolve the problems of the existing methods. The proposed method uses the 60-degree-pruning method [3]. The proposed method does not need to reprocess the query for continuous query processing because the proposed method draws and monitors the monitoring area including the candidate objects of a RkNN query.

2. The Proposed Method

Figure 1 shows the proposed RkNN query processing method. The proposed method consists of two phases such as local processing phase and global processing phase. First, in local processing phase, each object is assigned to a node randomly. Mapper computes the key and value of an object. Key means the section of objects. Value means the distance between objects and the query. And then, mapper sorts objects. In this step, mapper prunes objects that are not included in the result of RkNN such as P_2 and P_4. Second, in global processing phase, each object is assigned to a node by key. And then, mapper prunes objects that are not included in the result of RkNN such as P_3. Finally, reducer merges the results of RkNN. As a result, six candidate objects remain. They are the closest object from the query in each section such as P_1, P_5, and P_{15}. Finally, the proposed method computes the result of RkNN through the verification of candidate objects.

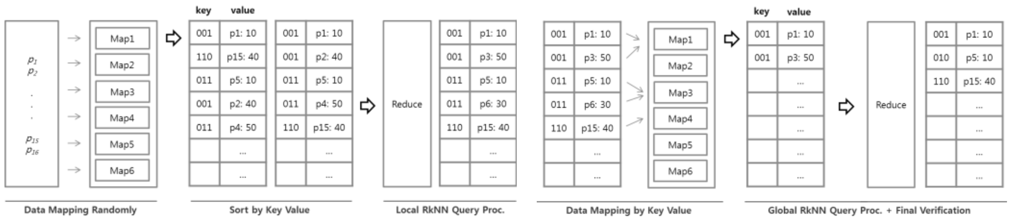

Figure 1. The Proposed RkNN Query Processing Method

Figure 2 shows the proposed continuous RkNN query processing method. In order to process the continuous RkNN, the proposed method draws the monitoring region as shown in Figure 2. The monitoring region is drawn by

* corresponding author : yjs@chungbuk.ac.kr
This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1022) supervised by the NIPA(National IT Industry Promotion Agency), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No.2013R1A2A2A01015710), and the Ministry of Education (MOE) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation (No.2013H1B8A2032298)

candidate objects. Mapper updates the locations of objects and monitors the monitoring region of each section. When a certain object intersects the monitoring region such as P_{14}, the proposed method updates a candidate object list. Finally, the proposed method updates the result of RkNN query using only the global processing phase.

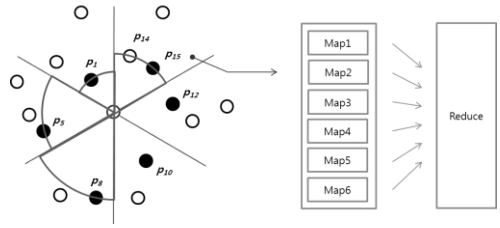

Figure 2. The Proposed Continuous RkNN Query Processing Method

3. Performance Evaluation

In order to show the superiority of the proposed method, we compared it with the query processing performance of the existing method [2]. All experiments have been performed on a Windows 7 operating system with a 3.0 GHz CPU and 2 GB main memory. In each experiment, we performed reverse k nearest neighbor queries in the particular data set that has 1K objects generated in the 1,024 ×1,024 space.

Figure 3 shows the number of object accesses according to the number of RkNN queries. As the number of queries increases, the query processing costs of the existing method and the proposed method are increased. In this experiment, the existing method archives better performance than the proposed method because this experiment is executed in uniform (randomly generated) environments. The existing method has about 4~6 candidate objects in the experiment of uniform environments. However, the proposed method has 6 candidate objects in the experiment. In other words, the pruning method of the existing method achieves better performance than the proposed method in uniform environments. However, the pruning method of the proposed method achieves much better performance than the existing method in non-uniform environments.

Figure 4 shows the number of objects accesses according to the number of continuous steps. As the number of continuous steps increases, the query costs of the existing method and the proposed method are also increased. As a result, the proposed method achieves about 256% better performance than the existing method at continuous query processing because the existing method must redraw bisectors to process the continuous query. However, the proposed method does not need to reprocess the query for continuous query processing because it draws and monitors the monitoring area including the candidate objects of the RkNN query.

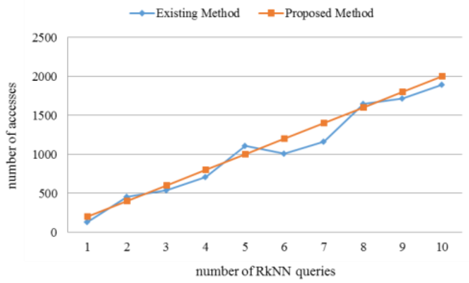

Figure 3. Performance according to the Number of RkNN Queries

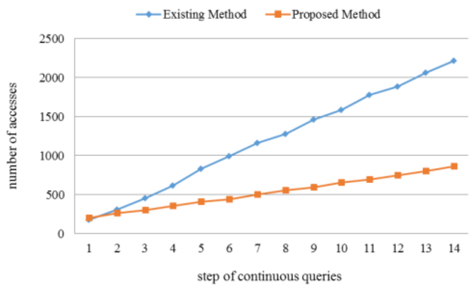

Figure 4. Performance according to the Number of Continuous Steps

4. Conclusion

In this paper, we proposed an efficient continuous RkNN query processing method with MR to solve the problems of the existing methods. The proposed method uses the 60-degree-pruning method. Since the proposed method draws and monitors the monitoring area including candidate objects of the RkNN query, it does not need to reprocess the continuous query. In order to show the superiority of the proposed method, we compared it with the existing method in terms of the query processing performance. As a result, the proposed method achieved about 256% better performance than the existing method in terms of the number of data accesses.

5. References

[1] A. Akdogan, U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, “Voronoi-based Geospatial Query Processing with MapReduce”, Proc. of the IEEE Cloud Computing, 2010.
[2] C. Ji, H. Hu, Y. Xu, and Y. Li, “Efficient Multi-dimensional Spatial RkNN Query Processing with MapReduce”, Proc. of the ChinaGrid Conference, 2013.
[3] I. Stanoi, D. Agrawal, A. E. Abbadi, “Reverse Nearest Neighbor Queries for Dynamic Databases”, Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000.